

GSOC SoE-Editor 2.0 - A Generic Sequence of Events Tool

Jens Hartung* Rainer Nibler* Andreas Spörl† Christoph Lenzen*

Maria Theresia Wörle‡

*Deutsches Zentrum für Luft- und Raumfahrt e.V., German Aerospace Center,
Münchener Straße 20, 82234 Weßling, Germany*

Chris Peat§

Heavens-Above GmbH, 81377 München, Germany

At the German Space Operations Center (GSOC) two applications had been developed for scheduling operations of Launch and Early Orbit Phases (LEOP), Commissioning Phases or special operations campaigns (e.g. software upload, special orbit maneuvers, etc.): one for low earth orbit (LEO) and one for medium (MEO) and geostationary earth orbit (GEO). The experiences of these tools were now merged with the scheduling capabilities of GSOC's generic mission planning application Pinta (Program for interactive timeline analysis), its scheduling library Plato (Planning tool) and the GSOC web based timeline display TimOnWeb.

Nomenclature

<i>FOP</i>	Flight Operation Procedure
<i>GEO</i>	Geostationary Earth Orbit
<i>GOP</i>	Ground Operation Procedure
<i>GSOC</i>	German Space Operations Center
<i>GUI</i>	Graphical User Interface
<i>JSON</i>	JavaScript Object Notation
<i>LEO</i>	Low Earth Orbit
<i>LEOP</i>	Launch and Early Orbit Phase
<i>MEO</i>	Medium Earth Orbit
<i>MET</i>	Mission Elapsed Time
<i>ORI</i>	Orbit Related Information
<i>Pinta</i>	Program for interactive timeline analysis
<i>Plato</i>	Planning tool
<i>SoE</i>	Sequence of Events

*Mission Planning System Engineer, Mission Operations Department, German Space Operations Center

†Deputy Head of Low Earth Orbit Spacecraft Operations Team, Mission Operations Department, German Space Operations Center

‡Deputy Head of Mission Planning Team, Mission Operations Department, German Space Operations Center

§Head of Heavens-Above GmbH

I. Introduction

The generation of Sequences of Events (SoEs) has a long tradition at GSOC, as it launched and positioned various geostationary spacecraft for commercial customers since the mid 1970s.¹ The development of the first GSOC SoE-Generator started in the 1990s.² Since then a few iterations of the initial SoE-Generator had been made including the development of a completely new SoE-Editor for the launch of the TanDEM-X spacecraft in 2010 and following LEO missions. The latest versions of the existing two SoE-Editors will be described shortly in the following subsections.

A. Overview of the existing MEO/GEO SoE-Editor

The oldest of the existing SoE editors is the one written for MEO and GEO missions. This tool imports Orbit Related Information (ORI) and Azimuth, Elevation, Range (AER) files provided by the GSOC Flight Dynamics Department.³ The ORI file contains orbit events related to the spacecraft, e.g. orbit number, elevation over a ground station, eclipse phases, etc. These are needed to have reference points for scheduling the SoE. From the AER file, which includes information for the ground stations to track the spacecraft, mainly the elevations are read to have visual elevation profiles for each ground station in the graphical SoE view.

Another input for the SoE editor are the Flight Operation Procedures (FOPs) and Ground Operation Procedures (GOPs), which will be performed during the execution of the SoE. These have to be entered manually to a Microsoft Access database, which will then be used by the SoE-Editor to store all SoE relevant information (including the imported Flight Dynamics Events and the final SoE). The user has to then create the SoE by adding time labels and relations directly within the Access database. Only the scheduling of ground station supports and the needed manpower for SoE execution can additionally be done within the graphical SoE view provided by the SoE-Editor.

During the execution of the SoE, the user has the possibility to import updated Flight Dynamics input files. The SoE-Editor then recalculates the SoE considering the relations specified during the SoE generation. The user then has to manually check if the SoE is still valid as constraint checks (e.g. double ground station visibility for critical events) cannot be specified in the SoE-Editor. Other changes within the SoE have to be done in the same way as it was done during the generation.

For documentation purposes the SoE-Editor supports to generate PDF files including the SoE in a tabular style where each event has a single line containing the times in UTC and MET and additional information such as ground station visibilities or active orbit constraints. Additionally, the SoE can also be exported in a PDF file containing the SoE in a graphical way, which is similar to the graphical view of the SoE (see Fig. 1, the top part with the alphanumeric table is missing for the export).

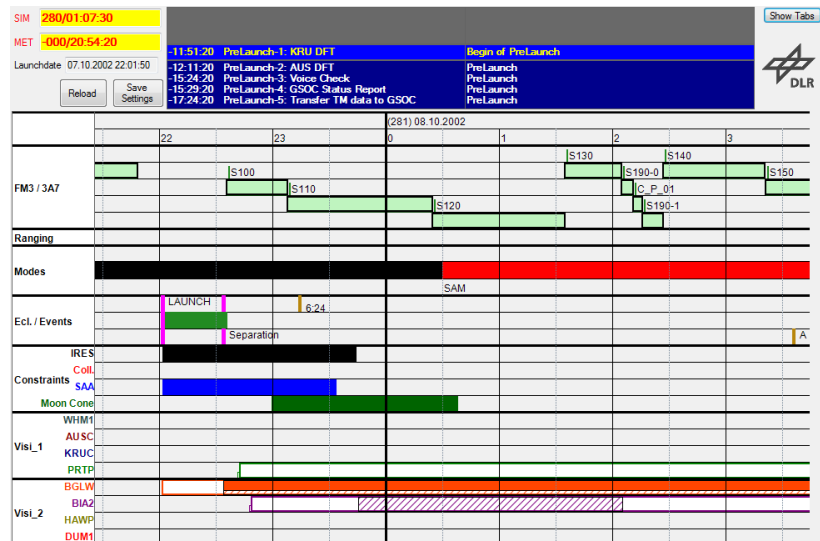


Figure 1. Graphical view of the existing MEO/GEO SoE-Editor containing an alphanumeric table at the top followed by plots which show the FOPs and GOPs, Attitude and Orbit Control System (AOCS) Modes, special events (such as Launch and Separation), short lasting orbit events, longer lasting orbit events and the ground station visibilities.

B. Overview of the existing LEO SoE-Editor

Prior to the development of the LEO SoE-Editor, the SoE for a LEOP of a LEO spacecraft was mainly built within a Microsoft Excel table, which would have to be imported to and exported from the planning

tool used in former times. As these processes didn't work properly due to necessary translations between an object based data model containing many relations and a tabular data model without relations, the decision was made to develop a web-based application for the SoE generation. This SoE-Editor had the requirement to allow the easy generation of a SoE without losing the tabular character, which was well-rehearsed at that time.

File	Edit	View	New	Numbering	Times	Replace	Misc	Execution Timeline
Undo	Redo	Select all	Invert Selection	Clear Selection	Duplicate selected	Delete selected	Number selected	

Page 1 of 20

first previous next last

Page Size: 100

1 2 3 4 - 18 19 20

	Row#	Time/UTC	Current UTC	SimTime/UTC	Duration	Station	ProcID/Upload	ProcID/Execution	Procedure Title	Activity	Status/Changes	Remarks	Execution Notes	Selection	Shift	State
Repeat Columns	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<
	15	MET -00:41:04	2012-204706:52:152	2012-187708:52:152	00:00:30					Lift-off date and time from Bankbook						
	16	MET -00:08:00	2012-204707:25:192	2012-187707:25:192	00:00:30					Injection into separation orbit						
	17	MET -00:06:30	2012-204707:26:482	2012-187707:26:482	00:00:30					Separation of BKA S/C						
	18	MET -00:01:30	2012-204707:31:482	2012-187707:31:482	00:00:30					Separation of Kanopus-B S/C						
	19	MET	2012-204707:33:192	2012-187707:33:192	00:00:30					Separation of TET-1 S/C						
	20	2012-204708:23:44.900Z	2012-204708:23:442	2012-187708:23:442	00:09:23	SP2BG-5				SP2BG-5						TET
	21	2012-204708:23:45.900Z	2012-204708:23:452	2012-187708:23:452	00:00:00	WHM1-S				WHM1-S						TET
	22	2012-204708:24:41.100Z	2012-204708:24:412	2012-187708:24:412	00:07:30	SP2BG-10				SP2BG-10						TET
	23	MET +1 PASS-10 +00:03:00 #1	2012-204708:27:412	2012-187708:27:412	00:01:00			TET_COM3010T	TET_COM_10_TEST_CMDS							
	24	MET +1 PASS-10 +00:03:40	2012-204708:28:212	2012-187708:28:212	00:01:00			TET_COM1000N	TET_COM_ADMODE_INIT_F00VR							
	25	MET +1 PASS-10 +00:04:00	2012-204708:28:412	2012-187708:28:412	00:01:00			TET_COM3010T	TET_COM_10_TEST_CMDS							
	26	MET +1 PASS-10 +00:05:00 #1	2012-204708:29:412	2012-187708:29:412	00:01:00			Ri-270 TET_ACS1300C	TET_ACS_EPC_RESET							
	27	MET +1 PASS-10 +00:05:30 #1	2012-204708:29:412	2012-187708:29:412	00:01:00			TET_SYS0400N	SYS_NOMINAL_PASS_EHND_CHECK							
	28	MET +1 PASS-10 +00:05:30 #1	2012-204708:30:112	2012-187708:30:112	00:01:00			Ri-271 TET_ACS1008C	TET_ACS_SFIM_MODE							
	29	MET +1 PASS-10 +00:06:00 #3	2012-204708:30:412	2012-187708:30:412	00:01:00			Ri-272 TET_ACS1001N	TET_ACS_AAM_MODE							
	30	2012-204708:32:11.900Z	2012-204708:32:112	2012-187708:32:112	00:00:00	SP2BG-10										TET
	31	2012-204708:33:08.300Z	2012-204708:33:082	2012-187708:33:082	00:00:00	SP2BG-5										TET

Figure 2. Editing view of the existing LEO SoE-Editor, where the user can add new SoE activities and modify existing ones.

The SoE-Editor operates within a web browser and initially displays the possible ground station passes which got imported from an ORI file provided by the GSOC Flight Dynamics Department. The user then has the possibility to include or exclude the imported ground station passes for the SoE generation since not all possible passes are needed or available during the LEOP. User-defined events containing an activity description, remarks and other information can be scheduled relative to the included ground station passes, MET or UTC.

The FOPs to be performed during the SoE execution can be allocated to these events via a drop-down menu, which was filled by reading a text file containing the procedure identifiers (which are unique identifiers for each FOP of the spacecraft). It is possible to add recurring events (like pre- and post-pass briefings) by defining template files, which are read by the SoE-Editor and then added to the SoE according the time constraints specified in the template.

Necessary maintenance activities of the SoE during the LEOP like removal of failed ground station passes and the addition of replacements are easy, when the assigned events are allowed to be moved to the following ground station passes in sequence. When they have to be moved to different or to later than the following passes, the user has to be firm with the ground station pass numbering and the time reference definitions. Bigger changes for the ground station passes (e.g. in case of a failed maneuver) may also be tricky for the SoE maintenance, as the user won't get a sufficient feedback during the reimport of the according Flight Dynamics file.

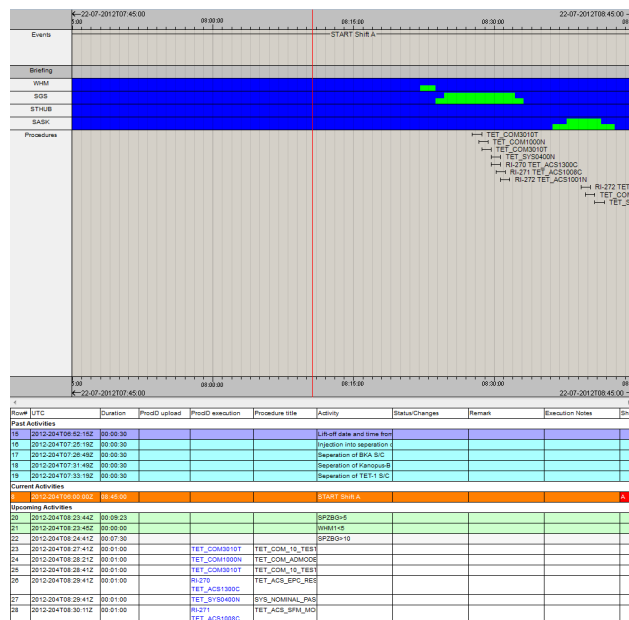


Figure 3. Display of LEO SoE in external application called TimOnWeb. In the graphical part at the top the events, FOPs and possible ground station passes will be plotted while at the bottom the events will be listed in an alphanumeric table (categorized by past, current and upcoming activities).

To improve the acceptance of the newly developed SoE-Editor it is possible to export the SoE as a Microsoft Excel file, which can then be adapted by the spacecraft system engineers. Also, the import of Excel files is possible but the implementation of this feature was easier this time, as the SoE is also stored as a table. Other possible export formats are a tabular-separated values (TSV) file or a printable web page containing the SoE table as seen within the SoE-Editor (see Fig. 2), but without the additional user interface elements at the top and left.

For displaying the SoE (e.g. on the big screens within the control room) a separate web application called TimOnWeb (see Fig. 3) will be used. Therefore, the SoE will be exported in the data model used by the application Pinta (Program for interactive timeline analysis)⁴ as this data model can be loaded by TimOnWeb. The graphical view for TimOnWeb will also be built in Pinta as the graphical display definition file is also the same for Pinta and TimOnWeb.

II. Motivation and Design for the new SoE-Editor

In 2011 and 2012 during preparation and execution of the LEOP for the TET-1 spacecraft usability improvements within the LEO SoE-Editor⁵ and new instabilities in the data exchange between the SoE-Editor and the timeline display application TimOnWeb were discovered. At the same time, requirements for new GEO spacecraft missions were defined indicating necessary improvements for the existing GEO SoE-Editor. Furthermore the maintaining of two different tools for the same purpose lead to the consideration to develop a SoE-Editor for all missions (independent of spacecraft type), which combines the advantages of both existing tools and eliminates the disadvantages. Therefore the following top level requirements had been defined:

- Possibility to (re-)import FOPs and GOPs
- Possibility to (re-)import orbit related events for spacecraft
- Possibility to (re-)import mission specific events without having to adapt the core application
- Easy generation of the SoE within a graphical timeline display
 - Adding FOPs/GOPs and custom defined events via Drag&Drop to the timeline
 - Moving/Deleting of non orbit related events directly within timeline display
 - Definition/Modification/Deletion of time dependencies between events directly within the timeline display
- The workload for updating the SoE after a launch delay shall be as low as possible
- Possibility to display finished SoE on big screen environment in control rooms and on local screens
- Possibility to export the generated SoE to different file formats (mainly PDF and CSV)

Different well-proven mission planning applications have been analysed if they were able to fulfill the requirements and can be improved with reasonable effort. The GSOC's own generic mission planning application Pinta was quickly selected as an adequate candidate and therefore was used as a base for the new SoE-Editor. Pinta has a powerful graphical timeline display for creating, modifying and viewing mission timelines. The events in Pinta are stored in the flexible GSOC planning modelling language⁶, which allows to store events in different groups for distinction purposes. Pinta also has an interface that has full access to the data model allowing to import, modify and export SoE related data. As described in the presentation of the LEO SoE-Editor, projects created and stored in the Pinta data model can be displayed in an external web application called TimOnWeb, which can be used for showing the SoE on both big screen environments and local screens.

Despite the substantial base Pinta and the existing TimOnWeb have some disadvantages and missing several features which have to be considered and improved. These shall be described in the following two sections (separated by Pinta and TimOnWeb):

III. New Features and Improvements for Pinta

Most of the possibilities provided by Pinta can be used without adaptations for the new SoE-Editor, e.g. the plug-in interface can be used for generation of events that are specific to single missions. But as for many other well-proven applications new features and improvements are needed to use them for a new use case. Four of the necessary extensions and enhancements shall be described in detail in the following subsections.

A. Extension of the GSOC Planning Modelling Language

The GSOC planning modelling language fulfills main requirements for the new SoE-Editor as tasks can be defined for all necessary events needed to build a SoE and allows to pool them in groups. Also the possibility to create time dependencies between tasks, which are needed to define time relations between events, is a key feature needed for the new SoE-Editor. But these dependencies have one main disadvantage as they are modeled between two tasks, which can have multiple timeline entries (instances of the task) meaning the time dependencies defined for the tasks have to be considered for each timeline entry of the depending task (see Fig. 4).

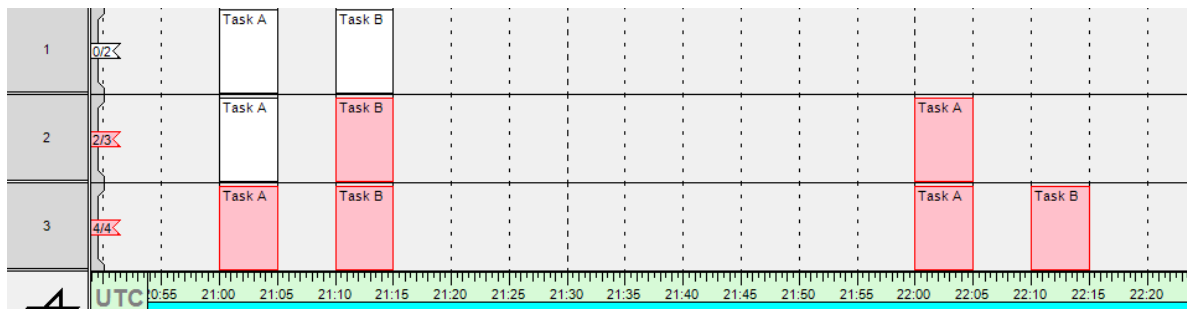


Figure 4. The issue existing for time dependencies when a task has multiple timeline entries is shown. For this example two tasks (Task A and Task B) were defined which have the time dependency, specifying that the start of Task B has to be exactly five minutes after the end of Task A. Row 1 shows how the time dependencies are designed to work. In row 2 an additional timeline entry for Task A was added to the timeline exactly one hour after the first timeline entry of Task A. This one immediately shows a conflict as the time dependency to Task B is violated (it starts before the start of second timeline entry of Task A). The timeline entry of Task B also displays a conflict because the time dependency between both tasks have to be valid for all timeline entries of each task. Trying to solve the conflict by adding a second timeline entry for Task B five minutes after end of the second timeline entry of Task A leads to all timeline entries of both tasks having a conflict (see row 3) due to the last-mentioned fact.

This issue would lead to a higher workload during the SoE generation, for example when a procedure sequence between multiple FOPs has to be defined and one or multiple of the procedures are part of more than one sequence as the user must ensure that these procedures only have one timeline entry per task (meaning he has to copy the procedures). But copying procedure tasks would have the side effect that the reimport of procedure will get more complex, as all copies have to be found and updated accordingly. These facts lead to the decision to extend the GSOC planning modelling language for the SoE-Editor by adding two new types of tasks: first the Template Task for which no timeline entries are allowed and are the base for the second new task type, the Instance Tasks. They are specified by having a relationship to the assigned Template Task (via a task parameter) and having at most one timeline entry per Instance Task. The properties of the Instance Tasks cannot be modified as they are derived from the template.

Translated to FOPs: the procedure will be imported as a Template Task. When the user wants to add an instance of this task to the SoE timeline via Drag&Drop, an Instance Task will automatically be created for the procedure. If the timeline entry of an

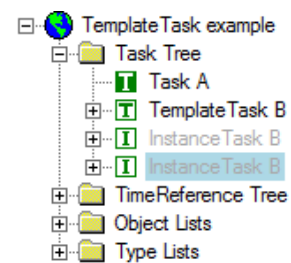


Figure 5. The additional task types seen in the Pinta Project Tree. The first task (Task A) is a normal one as defined previously in the GSOC planning modelling language. The second task (TemplateTask B) is defined as a Template Task. These tasks can be distinguished by the inverted icon and the prefix 'TemplateTask'. The last two tasks are Instance Tasks derived from the previous Template Task. They have an 'I' within the icon instead of a 'T' and have the prefix 'InstanceTask'.

Instance Task within the graphical timeline view will be copied, an entire new Instance Task will be generated again (without losing the relationship to the Template Task). Changing the properties of the Template Task (e.g. due to a reimport of FOPs) leads to the update of all assigned Instance Tasks (via the relationship stored in the task parameter). The creation of time dependencies between multiple events of same type is much safer with this extension than without it.

B. New importer for Flight and Ground Operation Procedures

Both of the existing SoE-Editors have the disadvantage that FOPs and GOPs have to be defined manually before they can be used during the generation of the SoE. Currently, operational used FOPs at GSOC will be created and maintained with the software MOIS (Manufacturing and Operations Information System)⁷ developed by RHEA. This software allows to export the Flight Operations Procedures as XML files, which can be validated against a provided XML schema file. These files are suitable for being parsed by an importer, which creates Template Tasks introduced in previous subsection for each exported procedure and as stated in the Design definition a plug-in implementing the Pinta data model interface is ideal for importing and exporting data to/from the Pinta data model.

Before developing the importer plug-in a detailed view into the FOP definition should be made to figure out the data structures to be created. A FOP mostly contains: the procedure identifier, procedure title, description, procedure duration, and one or multiple steps. Examples for that steps are spacecraft telemetry checks, telecommand statements, and other procedure execution instructions. Also branches and loops are possible for sequencing the steps. For GEO missions, where one FOP can last multiple hours (as they have near everytime visibility opportunities to ground stations), these steps may be important to be also displayed within the SoE, so they have to be imported where possible. In contrast they are mostly not necessary for LEO missions, as the procedures last only a few minutes and mostly don't include advanced instructions like decisions or loops. Therefore, the importer shall read the base information of the procedure while the detailed instructions defined within the steps of the procedure shall only be imported on request. When importing the steps of the procedure, the importer has to create additional tasks and link them to the procedure task or the tasks of previous steps via time dependencies. Otherwise the SoE-Editor doesn't have the possibility to schedule the steps to their correct locations within the SoE.

Another fact that should be considered during the development of the FOPs importer is the execution of a procedure. Especially for LEO missions they can be commanded in real-time or can be uploaded earlier to an execution list and are triggered at a predefined time (time-tagged). When modeling this during the SoE generation, the user usually would like to have the ability to define the uplink time of a time-tagged procedure. Therefore, an additional task has to be available for these procedures representing their uplink opportunity.

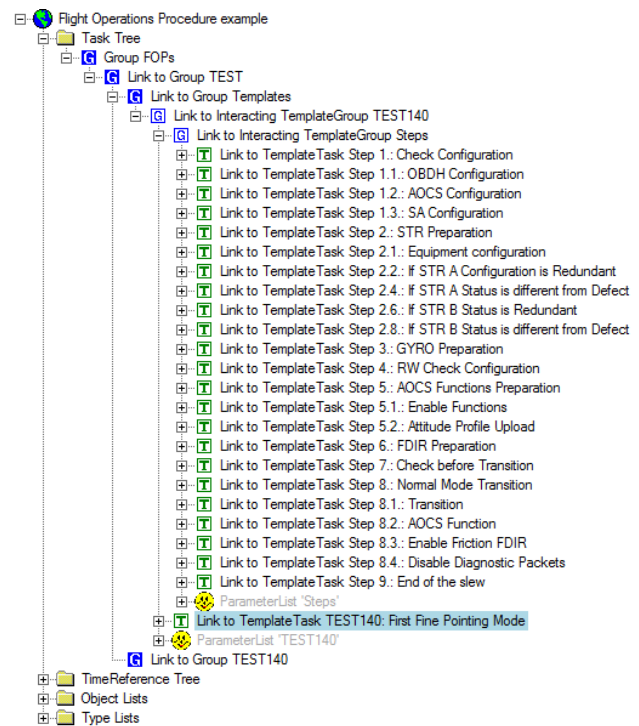


Figure 6. Example of an imported procedure within the Pinta Project Tree. The FOPs are created within a top-level group 'FOPs' which have the spacecraft identifier as subgroup (as one procedure may be valid for multiple spacecraft). Within this group, the procedure templates are stored in a 'Templates' group while the instances are located in a group representing the procedure identifier. This identifier will also be used for pooling the tasks created for the procedure within the Template. Then the procedure itself is stored as a task, while the procedure steps (each step is a separate task) are again combined in a 'Steps' group. The example procedure contains four decisions steps (Step 2.2, 2.4, 2.6 and 2.8) which have sub-steps that have not been imported due to complexity reasons.

C. Enhancement of the User Interface

Due to the complexity of the SoE-Editor and the huge number of constraints (especially time dependencies), the user has to deal with, it was necessary to improve the current Pinta GUI (Graphical User Interface) and provide an easy to grasp user interface for easy and fast SoE editing. Neither should the user be overstrained by confusing menus or cumbersome handling nor should any important information be hidden or hard to find. Furthermore the user must be capable to edit all relevant scheduling parameters directly from the user interface without any effort in the least amount of time. To achieve this goals several improvements to the existing Pinta GUI were necessary.

For generation of the SoE the user mainly has to perform two activities: adding events to the SoE and defining time dependencies among each other or to imported events. Since the most activities are FOPs and GOPs for a spacecraft LEOP SoE, a quick way for adding them to the timeline is needed. Pinta supports to Drag&Drop tasks from the Project Tree to the Timeline View, but the project tree has a main disadvantage: due to its much bigger scope of operation and consequently higher complexity a much simpler user interface shall be developed for the SoE-Editor to schedule FOPs and GOPs. The plug-in 'Tasks to schedule' is a reduced Pinta Project Tree like view (see Fig. 7) which gives the user the ability to find the Template Tasks for FOPs and GOPs without much effort and schedule them via Drag&Drop. As mentioned in subsection III.A new Instance Tasks are created based on the imported Template Tasks automatically including all procedure steps and sub-steps with the time dependencies between them and the Procedure Task (when imported). Because of the simple usability of the plugin the whole complexness is hidden for the user. Even an automatic repair is triggered after dropping a FOP or GOP on the timeline which ensures that the procedure is scheduled completely without conflicts.

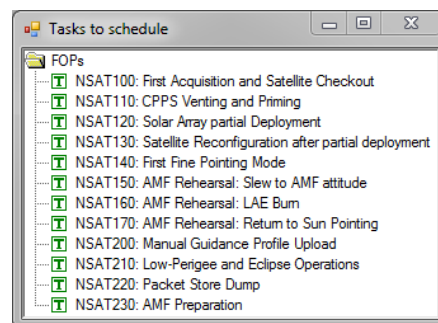


Figure 7. Example of the new 'Tasks to schedule' dialog used for an easy addition of FOPs and GOPs to the timeline. The procedure tasks shown in this dialog are only the imported Template Tasks, not the instances that will be actually added.

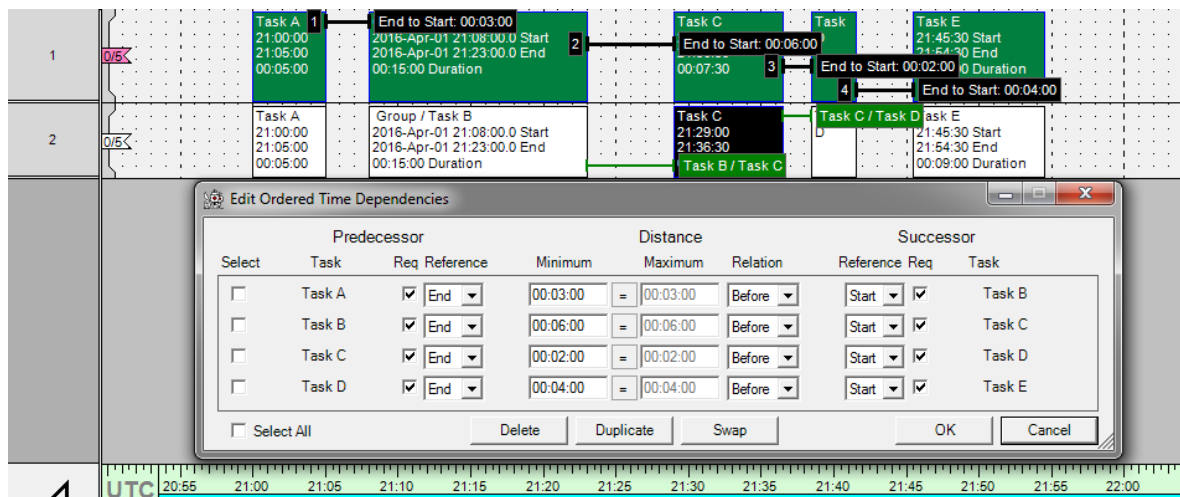


Figure 8. The developed improvements for creating and showing time dependencies between several tasks can be seen. In row 1 the 'Show Distances' functionality is active in the 'chain' mode. All tasks within the row have been selected starting from Task A and ending with Task E. The black lines between the tasks represent the distances displayed in the information boxes plotted at the right of them. Calling the operation to create time dependencies (via short-cut or the context menu of one of the information boxes) opens the 'Edit Ordered Time Dependencies' dialog, in which the user can adjust the distances between the tasks, change the reference anchor for each task (which determines if the dependency shall reference the start or end of the timeline entry) and specify if the opposite task of the dependency has to be available (a task with a time dependency to another task won't get a conflict, when the other task has no timeline entry scheduled - this option prevents this by adding an additional constraint) before creating the time dependencies. Row 2 shows the 'Show Ordered Time Dependencies' functionality. By selecting one or multiple timeline entries, the active time dependencies for these will be displayed. The user can modify and delete them in the 'Edit Ordered Time Dependencies' dialog.

Figure 8 demonstrates the solution for the second main task a user would like to do during the SoE creation, the definition of time dependencies between events added by the user himself and imported events. By simply selecting several timeline entries on the timeline - with the new developed 'Show Distances' tool enabled - he can create a chain, a ring or any kind of constellation of time dependencies between the tasks that are allocated to the timeline entries. An additional form called 'Create Time Dependencies' provides a more detailed editing of these constraints. The 'Show Ordered Time Dependencies' tool can be used to visualize already defined time dependencies with noticeable conflict status colors. Selecting these controls allows further modification or deletion.

During the execution of the SoE (in the LEO or a simulation run) some imported events may be shifted after a reimport, for example due to updated orbit parameters. In case time dependencies are assigned to these events, induced conflicts will appear indicating constraint violations. It has been proven to be a very tough problem for the user to resolve conflicts manually. Therefore, a new repair algorithm was developed (see Fig. 9) which can be triggered by simply clicking on the task causing the conflicts or calling it manually when the quick way is deactivated (to prevent false clicks). If a conflict-free and unique solution exists all timeline entries of tasks that are connected with time dependencies are adapted automatically in a recursive manner. If several conflict-free scenarios are possible, the first calculated solution is used and a warning is displayed to the user. Invalid or unresolvable time dependency connections are detected and no repairing is performed.

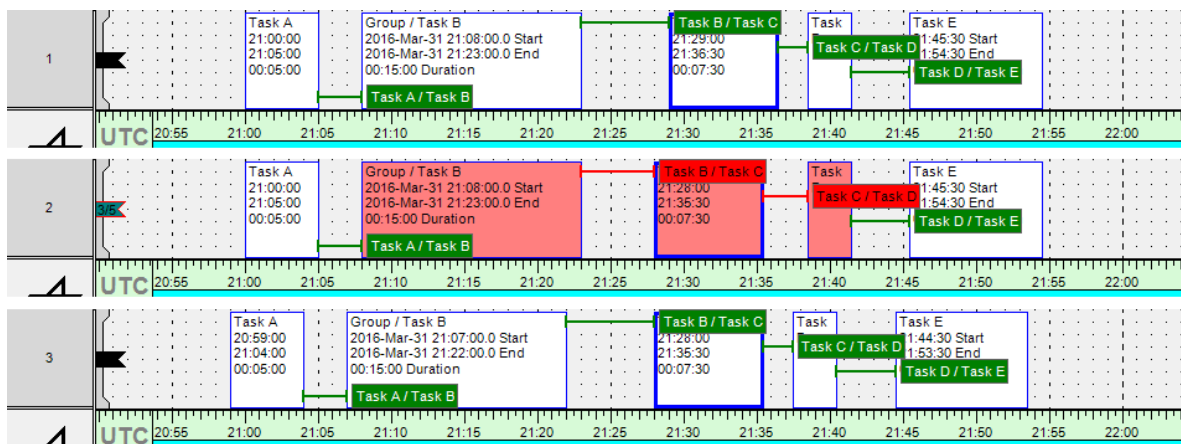


Figure 9. The repair algorithm for fixing conflicts related to time dependencies is demonstrated. Displayed are five tasks, Task A to Task E, that are chained via time dependencies. Task C shall represent an orbit related event (indicated by a wider border) which will be shifted after a reimport of Orbit Related Information. The first figure (with row 1) shows the initial constellation before the reimport is done. With the changed input the orbit related event assigned to Task C starts one minute earlier. Therefore, Task C will be shifted accordingly during the reimport. The second figure (with row 2) now indicates three conflicting timeline entries and two conflicting time dependencies. The user now can check the situation and make the decision that all timeline entries related to the shifted timeline entry of Task C can be moved, too. When the option to trigger the repair algorithm directly is active, the user only needs to click on the conflicting timeline entry of Task C to fix the sequence. Otherwise, he has to press an assigned short-cut or the according entry within the context menu of the timeline entry box to trigger the repair mechanism. The changes made by the algorithm can be seen in the last figure (with row 3).

D. Planning of Recurring Tasks

The existing LEO SoE-Editor supports the creation of recurring tasks, which can be assigned to specific events of the same type (e.g. Pre-Pass briefing ten minutes before each scheduled ground station pass). Pinta does not provide a similar feature meaning algorithms for the generation of according tasks would have to be added to the application source code directly or within a plug-in. But Pinta already has an interface to the GSOC Planning Tool Plato⁸ implemented, which has been used for similar problems before. Plato offers several planning algorithms, which can be configured via XML files containing the rules to be considered. One rule may be to schedule an event every 30 minutes but only when a specified resource is not filled at the considered time. If the resource is filled, it shall find the next opportunity where the resource is empty again, schedule the event and restart the algorithm recursively until a predefined time frame has been

considered. This example can be used for scheduling spacecraft ranging periods, which have to be performed regularly, but only when no FOP is executed in parallel (see Fig. 10).

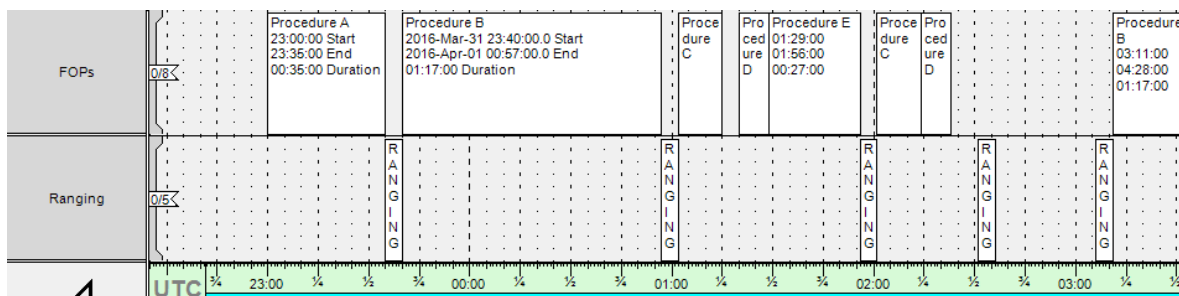


Figure 10. Example of a recurring task for which spacecraft ranging periods shall be scheduled every 30 minutes when there is no FOP in parallel. The first ranging task starts directly after the first procedure in the shown sequence at 23:35:00. The next opportunity would be 00:05:00, but at this time a FOP is active leading to scheduling the second ranging task after the end of this procedure (00:57:00). The algorithm will be restarted recursively so that the next opportunity will be 01:27:00. Thus, the gap at around 01:15:00 will be ignored. At 01:27:00 another procedure is performed so that the next opportunity will be the following free gap at 01:56:00. In the big gap after the following short-lasting procedures the algorithm is able to schedule two consecutive ranging periods.

IV. Displaying and Export of a SoE

The SoE can be generated and maintained by all the new features and improvements we have discussed in the previous section. Another key requirements for working with the resulting SoE is the possibility to display it within both the control and support rooms as well as being able to export it to various formats which are able to be printed or read by other applications (e.g. the software for scheduling necessary ground stations in the correct configurations). As mentioned in the overview of the existing LEO SoE-Editor and the Design definition, there is already a tool which can read the Pinta data model and display the content within a web browser, which can be used to show the SoE on big screens in the control room. But this version of TimOnWeb has one major disadvantage, which will be described in the next subsection, that leads to a completely new version of TimOnWeb. In this context, the exports have also be re-developed as the existing TimOnWeb version only supported a preformatted HTML page (which can also be printed). A download of this extract is only supported by using the mechanisms provided by the displaying web browser.

A. TimOnWeb

The main disadvantage of the existing TimOnWeb is the performance of the display, especially when scrolling through the timeline. The reason for this issue is the design of the tool, where the displayed content is rendered on the web server and the result is transferred to the client. For timelines containing a lot of information to be displayed, the size of the rendered images are too large for working with the timeline display without having to wait few seconds particularly when navigating to earlier or later times in the timeline (as the server has to render a new image for the new time period). The chosen architecture was state-of-the-art when the existing TimOnWeb was developed initially but is now quite out-dated with having new web browser techniques like HTML5 (supports rendering of images on client side) and WebGL. Both web browser features allow the reduction of workload for the web servers and most notably the amount of data needed to be transferred to the display clients. These facts lead to the decision to develop an entire new version of TimOnWeb using HTML5, JavaScript, JQuery and other libraries for client side rendering the graphical part of the timeline to be displayed. The data to be shown will be provided by the server via JSON (JavaScript Object Notation) files containing only the needed display settings and data objects.

Moving the generation of the graphical content of the SoE to the client, the main task for the server is to provide the necessary data as quickly as possible. To accomplish this, a separate conversion tool reads only the data needed to be displayed from the Pinta data model and serializes the result to a binary file for faster read access. The server then reads the binary file and caches the content within the local RAM (Random Access Memory). The display definition file (which specifies the contents to be displayed) has been rewritten

for the new TimOnWeb to support new display features like font size for textual elements in the graphical view or new plot types. This file is read by the server and will be transformed to a JSON file on initial request by the client. After the client has interpreted the display configuration, it will request the data to be shown in a specific time frame. The server then builds a JSON file containing all data matching to the filters by using the cached data pool and returns them to the client. As the JSON files have a size of few kilobytes even for greater amounts of data, the communication between client and server is quite fast.

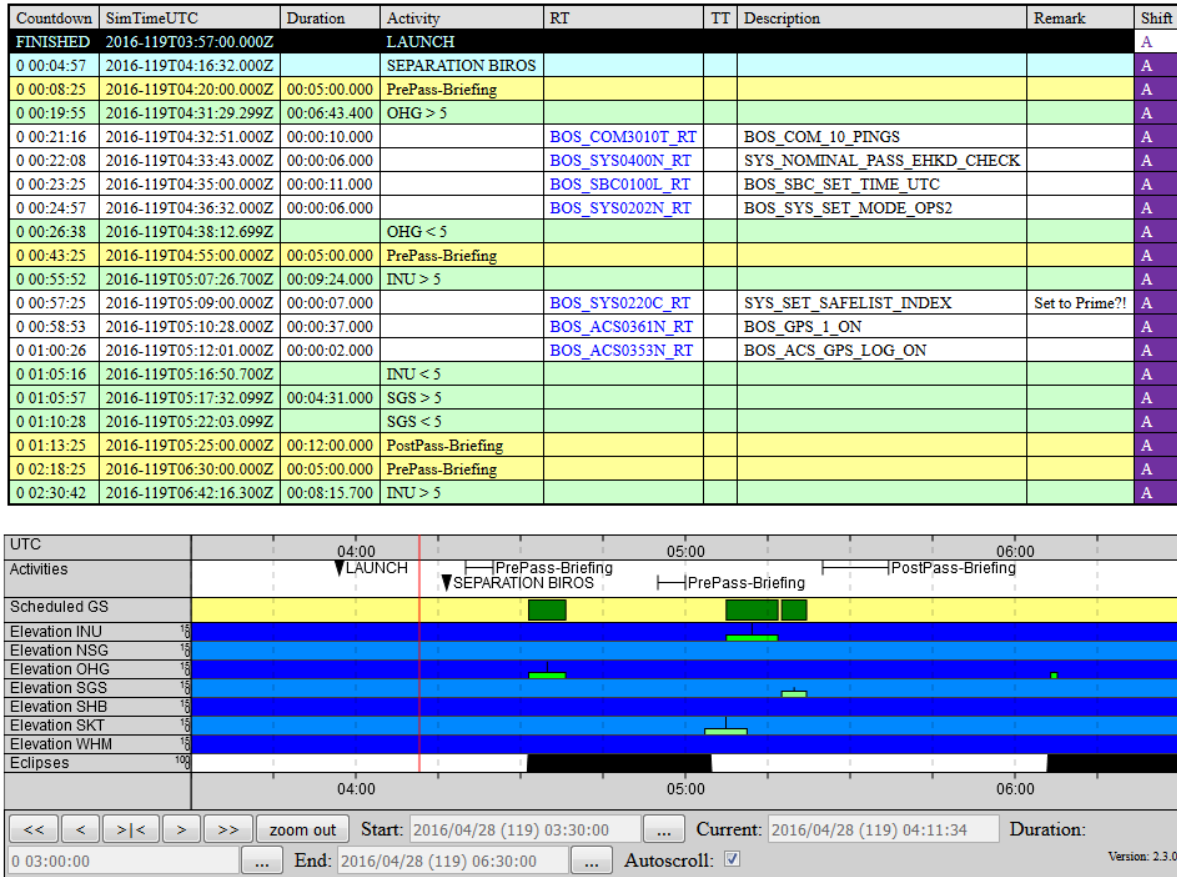


Figure 11. View of the new TimOnWeb displaying the SoE for the upcoming LEOP of a LEO spacecraft called BIROS. At the top the alphanumeric table can be seen showing past, currently active and upcoming events. In this example the table shows the activities of the first ground station passes after the separation of the spacecraft. The graphical part at the bottom presents parts of the events on a timeline. It contains additional events like the solar eclipses in the last plot or all possible ground station passes (not only the scheduled ones).

To support simulation runs based on a finished SoE prior to a launch, a separate web page is available on the server, where the user can set and modify time offsets that will be considered for displaying the live content (with scrolling as the time continues). The time offsets can be defined by first specifying a nearby date and secondly the wished target date. When the current time is at the nearby date or after that, the shown content in the live display jumps to the target date added by the offset the current time is behind the defined nearby date. It is possible to define multiple offsets to support jumps within the SoE during simulation runs.

For displaying the alphanumeric part containing the SoE activities in the shape of a table, the well-known HTML table will be used as it was done for the old version of TimOnWeb. For fetching the content to be shown within the table, another conversion tool will be used which will also be used for the exports described in the following subsection. The tool provides a JSON file, which will be read by the server to fill the content of the alphanumeric table.

B. PDF and CSV exports

As mentioned in the previous subsection, the necessary Pinta data will be serialized and stored in a binary file by a conversion tool for faster read access by the server. Together with the execution of this transformation other conversions can and shall take place in parallel to generate printable outputs or machine readable files that can be used by other applications and TimOnWeb itself (recall the alphanumeric table). For the export of the graphical view of TimOnWeb, the plotting techniques implemented for the clients have to also be implemented within the conversion tool as the server won't execute a web browser for drawing the images needed for the export. But for the creation of tabular exports (for PDF, CSV, JSON, etc.) other implementation methods have to be considered.

Biros - Sequence of Events (SoE)

#	CurrentUTC	Duration	Activity	RT	TT	Description	Remark	Shift
1	2016-119T03:30:00.000Z	08:30:00.000	Start of Shift A					A
2	2016-119T03:57:00.000Z		LAUNCH					A
3	2016-119T04:16:32.000Z		SEPARATION BIROS					A
4	2016-119T04:20:00.000Z	00:05:00.000	PrePass-Briefing					A
5	2016-119T04:31:29.299Z	00:06:43.400	OHG > 5					A
6	2016-119T04:32:51.000Z	00:00:10.000		BOS_COM3010T_RT		BOS_COM_10_PINGS		A
7	2016-119T04:33:43.000Z	00:00:06.000		BOS_SYS0400N_RT		SYS_NOMINAL_PASS_EHKD_CHECK		A
8	2016-119T04:35:00.000Z	00:00:11.000		BOS_SBC0100L_RT		BOS_SBC_SET_TIME_UTC		A
9	2016-119T04:36:32.000Z	00:00:06.000		BOS_SYS0202N_RT		BOS_SYS_SET_MODE_OPS2		A
10	2016-119T04:38:12.699Z		OHG < 5					A
11	2016-119T04:55:00.000Z	00:05:00.000	PrePass-Briefing					A
12	2016-119T05:07:26.700Z	00:09:24.000	INU > 5					A
13	2016-119T05:09:00.000Z	00:00:07.000		BOS_SYS0220C_RT		SYS_SET_SAFELIST_INDEX	Set to Prime?!	A
14	2016-119T05:10:28.000Z	00:00:37.000		BOS_ACS0361N_RT		BOS_GPS_1_ON		A
15	2016-119T05:12:01.000Z	00:00:02.000		BOS_ACS0353N_RT		BOS_ACS_GPS_LOG_ON		A
16	2016-119T05:16:50.700Z		INU < 5					A
17	2016-119T05:17:32.099Z	00:04:31.000	SGS > 5					A
18	2016-119T05:22:03.099Z		SGS < 5					A
19	2016-119T05:25:00.000Z	00:12:00.000	PostPass-Briefing					A
20	2016-119T06:30:00.000Z	00:05:00.000	PrePass-Briefing					A
21	2016-119T06:42:16.300Z	00:08:15.700	INU > 5					A
22	2016-119T06:45:42.000Z	00:00:05.000		BOS_ACS0301N_RT		BOS_ACS_OD_ON		A
23	2016-119T06:47:56.000Z	00:00:02.000		BOS_ACS0320N_RT		BOS_ACS_OP_ON		A
24	2016-119T06:50:32.000Z		INU < 5					A
25	2016-119T06:51:33.699Z	00:07:11.900	SGS > 5					A
26	2016-119T06:53:46.000Z	00:00:01.000		BOS_SBC4303N_RT		BOS_SBC_UPL_COMMAND_LISTX_CHECK		A
27	2016-119T06:55:00.000Z	00:00:01.000		BOS_SBC4303N_RT		BOS_SBC_UPL_COMMAND_LISTX_CHECK		A
28	2016-119T06:55:45.000Z	00:02:00.000	Uplink BGS				AppendPassive2Active	A
29	2016-119T06:56:00.000Z	00:00:01.000		BOS_SBC4303N_RT		BOS_SBC_UPL_COMMAND_LISTX_CHECK		A
30	2016-119T06:58:45.600Z		SGS < 5					A
31	2016-119T07:10:00.000Z	00:12:00.000	PostPass-Briefing					A
32	2016-119T08:15:00.000Z	00:05:00.000	PrePass-Briefing					A
33	2016-119T08:25:39.299Z	00:08:42.600	SGS > 5					A
34	2016-119T08:28:31.000Z	00:01:00.000		BOS_SYS0100L_RT		BOS_SYS_DEPLOY_SOLARP_N		A
35	2016-119T08:30:20.000Z	00:00:03.000		BOS_ACS1007C_RT		BOS_ACS_SPM_MODE		A
36	2016-119T08:31:03.000Z	00:00:07.000		BOS_ACS0403L_RT		BOS_ACS_SET_MOM_INERTIA		A
37	2016-119T08:31:50.000Z	00:00:02.000		BOS_ACS1101C_RT		BOS_ACS_ESTIMATOR_RESET		A
38	2016-119T08:32:33.000Z	00:00:05.000		BOS_ACS1001N_RT		BOS_ACS_AAM_MODE		A
39	2016-119T08:33:46.499Z	00:08:40.100	WHM > 5					A
40	2016-119T08:34:21.899Z		SGS < 5					A
41	2016-119T08:38:40.000Z			BOS_ACS0214N_TT		BOS_ACS_DPU1_ON	Insert in Active	A
42	2016-119T08:42:26.599Z		WHM < 5					A
43	2016-119T08:55:00.000Z	00:12:00.000	PostPass-Briefing					A
44	2016-119T09:20:00.000Z	00:00:10.000			BOS_ACS0214N_TT	BOS_ACS_DPU1_ON		A

Page 1 of 5

Generated On 2016.03.31 18:12

Figure 12. Excerpt from a tabular PDF export of a SoE. It shows the events as seen within Fig. 11 and the SoE content of a few following ground station passes. At the bottom of the table the upload (entry where procedure identifier in the 'RT' column is written in red color) and execution of a time-tagged procedure are scheduled.

First, it has to be defined which types of events shall be exported and shall these events get additional information from the data model that aren't directly allocated to them. One example: three shifts shall perform the SoE around-the-clock and in the export the responsible shift(s) shall be assigned as an additional information to each event in the table. This leads to the necessity to combine data for one entry, which doesn't have a direct relation. The Plato library again facilitates the implementation effort, as it provides functionalities to analyse resource states easily. It allows for example the easy location of specific resource fill states (which can be used to create a new event in the exported table) or to check the resource fill level at a specific event time (e.g. to assign a resource state to the event). Another important use case of the Plato library is the combination of multiple properties or parameters assigned to a task so that the information at different locations within the task can be merged easily (e.g. for adding the procedure title to the procedure identifier stored in another parameter). The generated events are stored within objects that are designed as a table (containing a table object, row objects and cell objects) which are the base for all file generators. The generated files can be included as download link to each web page on the server including the TimOnWeb pages or can be used for further content creation like the alphanumeric table above or below the graphical

view of the timeline.

V. Conclusion and Outlook

With the new features and enhancements for Pinta and the new TimOnWeb, the generic GSOC mission planning applications have been improved to be able to use them not only for routine operations as it was done previously but also for LEOP support. This is still extendable for further use cases. Not all of the mentioned parts have been completely implemented yet, but enough to use them already in the upcoming LEOP of the BIROS spacecraft⁹ planned for end of May 2016. The SoE has been created with the new SoE-Editor and will be maintained with it during the LEOP and the following Commissioning Phase. Accordingly, the first simulation runs were already performed with the new tools. We received overwhelmingly positive feedback from the operation engineers, especially for the new TimOnWeb. Next, the SoE-Editor and TimOnWeb will also be used for an upcoming GEO spacecraft scheduled to be launched end of 2016. In the future, all the other planned LEO and GEO missions operated at GSOC are designated to profit of the support by the new tool suite.

References

- ¹Brittinger, P., *Eleven Geostationary Satellites positioned by DLR-GSOC: Operations Management Methods*, 3rd International Symposium on Space Operations and Ground Data Systems, November 15-18, 1994, Greenbelt, Maryland, USA.
- ²Braun, A. and Wobbe, H., *The Sequence of Events Generator - An Integrated System to Assist Mission Operations*, 1st International Symposium on "Reducing the Cost of Spacecraft Ground Systems and Operations", September 27-29, 1995, Rutherford Appleton Laboratory, Chilton, Oxfordshire, United Kingdom.
- ³Kirschner, M., Herman, J. and Kahle, R., *Flight Dynamic Operations*, In: Spacecraft Operations, Pages 119-166, 2015, Springer-Verlag, Wien, ISBN: 978-3-7091-1802-3
- ⁴Spörl, A., Lenzen, C., Wörle, M. T., Hartung, J., Mrowka, F., Braun, A. and Wickler, M., *Mission Planning System for the TET-1 OnOrbit Verification Mission*, 13th International Conference on Space Operations (SpaceOps 2014), May 05-09, 2014, Pasadena, California, USA.
- ⁵Axmann, R., Gude, J. and Mühlbauer, P., *TET-1 SATELLITE OPERATIONS LESSONS LEARNED: PREPARATION OF MISSION, LEOP AND ROUTINE OPERATIONS OF 11 DIFFERENT EXPERIMENTS*, 62nd International Astronautical Congress, October 03-07, 2011, Cape Town, South Africa.
- ⁶*Planning Modelling Language*, URL: http://www.dlr.de/rb/en/Portaldata/38/Resources/dokumente/GSOC_dokumente/RB-MIB/GSOC_Modelling_Language.pdf [cited March 31, 2016].
- ⁷Heinen, W., Reid, S. and Varadarajulu, S., *Automation through On-Board Control Procedures: Operational Concepts and Tools*, 11th International Conference on Space Operations (SpaceOps 2010), April 25-30, 2010, Huntsville, Alabama, USA.
- ⁸Lenzen, C., Wörle, M. T., Mrowka, F., Spörl, A. and Klaehn, R., *The Algorithm Assembly of Plato*, 12th International Conference on Space Operations (SpaceOps 2012), June 11-15, 2012, Stockholm, Sweden.
- ⁹Wörle, M. T., Spörl, A., Hartung, J., Lenzen, C. and Mrowka, F., *The Mission Planning System for the Firebird Spacecraft Constellation*, 14th International Conference on Space Operations (SpaceOps 2016), May 16-20, 2016, Daejeon, Republic of Korea.